



GAMA 1.6: Advancing the art of complex agent-based modeling and simulation

Arnaud Grignard, Patrick Taillandier, Benoit Gaudou, Duc An Vo, Nghi Huynh Quang, Alexis Drogoul

► To cite this version:

Arnaud Grignard, Patrick Taillandier, Benoit Gaudou, Duc An Vo, Nghi Huynh Quang, et al.. GAMA 1.6: Advancing the art of complex agent-based modeling and simulation. Pacific Rim international Conference on Multi-Agents (PRIMA), Dec 2013, Dunedin, New Zealand. pp.117-131, 10.1007/978-3-642-44927-7_9 . hal-00932406

HAL Id: hal-00932406

<https://hal.science/hal-00932406>

Submitted on 17 Jan 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - ShareAlike| 4.0 International License

GAMA 1.6: Advancing the art of complex agent-based modeling and simulation

Arnaud Grignard¹, Patrick Taillandier³, Benoit Gaudou⁴, Duc An Vo², Nghi Quang Huynh⁵, Alexis Drogoul²

¹ UMI 209 UMMISCO/MSI, UPMC, France

² UMI 209 UMMISCO/MSI, IRD, Vietnam

³ UMR 6266 IDEES, CNRS/University of Rouen, France

⁴ UMR 5505 IRIT, CNRS/University of Toulouse, France

⁵ DREAM-CTU/IRD, CICT, Can Tho University, Vietnam

Abstract. Agent-based models tend to be more and more complex. In order to cope with this increase of complexity, powerful modeling and simulation tools are required. These last years have seen the development of several platforms dedicated to the development of agent-based models. While some of them are still limited to the development of simple models, others allow to develop rich and complex models. Among them, the GAMA modeling and simulation platform is aimed at supporting the design of spatialized, multiple-paradigms and multiple-scales models. Several papers have already introduced GAMA, notably in earlier PRIMA conferences, and we would like, in this paper, to introduce the new features provided by GAMA 1.6, the latest revision to date of the platform. In particular, we present its capabilities concerning the tight combination of 3D visualization, GIS data management, and multi-level modeling. In addition, we present some examples of real projects that rely on GAMA to develop complex models.

Keywords: Agent-based modeling, simulation, GIS, multi-level, ODE, platform, visualization, complex systems

1 Introduction

The last years have seen an important increase in the use of agent-based modeling (ABM) in various scientific and application domains. Some of these applications, that rely on large sets of data, are more and more demanding in terms of representation, simulation and interpretation of complex models. While the classical KISS [10][3] approach appeared to be well suited in the early years of ABM, this recent trend, which for instance includes developments in serious games, participatory approaches or integrated models, requires the ability to design and manage more descriptive and detailed models.

The KISS-based approach has given birth to a plethora of small-size, mostly theoretical, toy models that, although they are well suited for training purposes, have had the paradoxical effect of establishing ABM as a mainstream approach to

complex systems modeling while, at the same time, eliminating the reasons why people had wanted to adopt it in the first place. Building complex, incremental, data-driven modular models in NetLogo [28], for instance, is a difficult task. And it becomes even more difficult when different data sources, at different levels of representation, are to be used. Interpreting these models is also very tedious, given the lack of flexibility offered by the existing platforms in the visualization and parametrization of the models.

While some offers have tried to overcome these limitations (e.g. Repast [23], Mason [20]), they have failed until now to propose a credible alternative in terms of *modeling platforms*. Repast, for instance, is more a well designed *toolbox* than a *platform* as it requires modelers to be highly proficient in Java programming and Eclipse development, which is rarely the case in the scientific domains targeted by ABM. Similarly, these proposals also fall short in terms of *simulation platform*, as designing virtual experiments that combine easy parametrization and high-level interactive visualization is as complicated as building the model itself and requires the use of several external tools.

The GAMA⁶ modeling and simulation platform [9] [26], developed since 2007 as an open-source project, aims at overcoming these lacks by providing modelers - which are not, most of the time, computer scientists - with tools to develop and experiment highly complex models through a well-thought integration of agent-based programming, geographical data management, flexible visualization tools and multi-level representation.

GAMA provides a complete modeling language (GAma Modeling Language) and an integrated development environment that allows modelers to build models as quickly and easily as in NetLogo while going beyond what Repast or Mason offer in terms of simulated experiments. It is currently applied in several projects in environmental decision-support systems, urban design, water management, biological invasions, climate change adaptation or disaster mitigation. This paper aims at presenting the new features developed in the latest version of GAMA. We will focus in particular on the following features: the evolution of the GAML language for specifying models using multiple paradigms (especially combinations of mathematical and computer approaches), the seamless integration of geographical data and other spatial data, the tools offered to support high-level visualization of simulations and the multi-level description of models.

2 GAMA Meta-Model

The concepts and the operational semantic of GAML are completely described in a meta-model (see Figure 1) from which every model written in the language derives. This meta-model supports the development of multi-level agent-based models by considering three main sets of abstract classes that represent, respectively, the *entities*, the *space* and the *time* of a model.

⁶ <http://gama-platform.googlecode.com>

- The representation of *Entities* are described by two main classes:
 - **Agent** represents an individual entity of a simulation, which is conceptually similar to an object in an object-oriented language. It represents a simulated entity in the simulation.
 - **Population** represents a group of agents sharing the same structure and behavior in a simulation. The population is responsible for managing the agents it is composed of.
- The representation of the *space* of a model is tightly coupled with the previous set and supported by two main classes:
 - **Geometry** is, as its name implies, a geometrical shape (supported by the JTS library ⁷) that represents the embodiment of an agent in its *environment*. It is strictly linked with one and only one agent.
 - **Topology** is meant to represent this *environment*. It provides a referential where geometries are located, an optional organization imposed to these geometries and a set of operations to support perception and movements of agents. A topology is linked with one population.
- The representation of *time* uses, similarly, a class linked with Agent and another one linked with Population (**Scheduler** and **Scheduling Information**).

These three layers of representation (*entities*, *space* and *time*) are tied together by the last class of the meta-model: **Species**. A species, like a class in OOP, defines the attributes, geometries and behaviors common to all the agents of a population, but it also specifies the topology and scheduling of this population. More importantly, the *containment* relationships between species allow to describe hierarchical levels of agency in a very natural way, by considering nested species of a species S as prototypes for micro-agents of agents instance of S. This translates concretely into the possibility for these agents to be hosts of a set of populations, each of them being described by a micro-species of S.

GAMA entirely relies on this meta-model to create models. The first level of agency (the model itself) describes the global topology of the model, its basic scheduling, its parameters and global behaviors, and is the host of the populations of agents described by the species written by the modeler. Recursively, each agent can become the host of nested populations, as long as its species contains the (micro) species that describe them. All of these populations can be provided, through the definition of their species, with their own spatio-temporal scale, which are not necessarily constrained by the containment hierarchy (i.e. the topology of a population of micro-agents can for instance define boundaries larger than the one described in the population of its host). This definition of species also supports the description of shared contexts between macro and micro agents as a way to express the dynamic relationships and data transfer between levels. Additionally, the morphogenesis operation, not presented in the diagram of the Figure 1, allows agents to change species during the simulation, their current species and their target species are linked by a materialization relationship,

⁷ <http://sourceforge.net/projects/jts-topo-suite/>

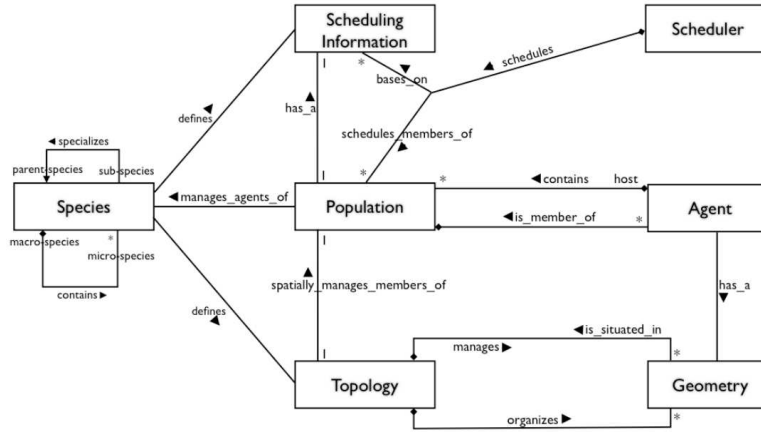


Fig. 1. GAMA meta-model

which effectively allows them to be represented at different levels of abstraction during a simulation.

3 GAMA features

3.1 The GAML modeling language

In order to ease the work of modelers, GAMA provides a complete IDE (Integrated Development Environment) supporting the definition of models. In particular, the IDE, based on Eclipse and XText, provides modern features such as auto-compilation, auto-completion, quick fixes, text coloring, etc. Modelers define their models using the GAML language. The interest of providing a DSL (Domain Specific Language) to modelers is to ease the definition of models. Indeed, the GAML language is very simple to use, even by modelers that have low programming skills.

GAML is an agent-oriented language. Modelers define species of agents, i.e. archetype of agents, their characteristics, behaviors and aspects. It is a typed language, variables (static or dynamic) has to be typed. The behaviors of agents are defined through actions and reflexes. An action is a block of instructions executed when the action is called. A reflex is a block of instructions executed at each simulation step or when its optional attached condition is true. An aspect represents how an agent can be displayed. It is possible to define as many aspects as necessary. The richness of GAML comes from the numerous optimized operators concerning geometries, containers, statistics and graphs.

Figure 2 shows an example of a simple epidemiological SI model definition. In this model, a species of agents called *people* is defined. This species has the moving *skill* which provides some built-in variables (speed, direction, etc.) as

well as some primitives(wander, goto, etc.). We redefine the variable *speed* in order to set its value to a random value between 5 and 10. The *people* species has another variable called *is_infected* that is a boolean initialized with true with a probability of 0.01. It has two reflexes: *move* and *infect*. The *move* reflex that is activated at each simulation cycle consists in a random movement: the agent uses the *wander* primitive of the moving skill to move to a random direction with a speed given by the value of the variable *speed*. The *infect* reflex is only activated when the agent is infected. It consists in trying to infect each people agent that is at distance lower or equals to 10m with a probability of 0.01. The *people* species has one aspect that will allow to display the *people* agents as a circle of radius 5m. The color will depend on the *is_infected* variable: if the agent is infected, its color will be red, green otherwise. At the initialization of the model, the world environment is initialized as a 500mx500m square and 1000 agents *people* are created (with a random location). The output of the *main_experiment* is a display called *map* where all the created agents can be visualized.

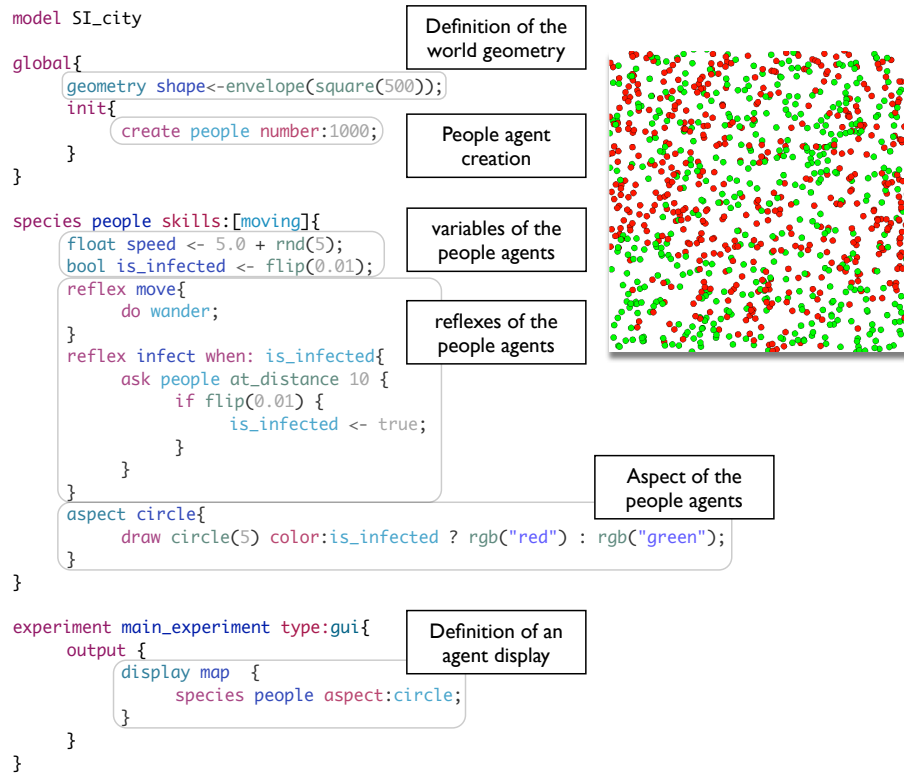


Fig. 2. Simple model definition in GAML and its display output.

3.2 Geographic information system integration

Geographic information system integration (GIS) is the digital representation of landscapes captured by cameras, digitizers or scanners. This technology can be used for scientific investigation, resource management and development planning. With properties such as location, attributes and spatial relationship GIS vocabulary is really closed to the one used in agent-based modeling.

Many recent models rely on GIS data, allowing them to gain in realism. If most of modern simulation platforms allow to read/write GIS data, not so many allow to agentify vector geometries. Indeed, if Netlogo[28] or Cormas [18] provide some basic primitives to load and manipulate GIS data, they do not allow to create agents with a complex geometry from GIS data even with their GIS plugins. Another limitation of most of modern platforms is that they do not allow to define several environments with different topologies. Typically, for some complex models, it is necessary to define different environments such as a continuous one, to integrate vector data or several raster data with different resolutions and eventually several graph environments representing different level of interaction within the model.

GAMA allows to define as many environments as necessary. It synchronizes all of them by using a continuous reference environment. It provides as well a geometry to each agent. This geometry, which is based on vector representation, can be simple (point, polyline or polygon) or complex (composed of several sub-geometries). It can be defined by the modeler (by using a list of points or pre-defined graphical primitives) or directly loaded from a shapefile, a raster file or a mesh file. Indeed, GAMA allows to use geographic vector data to create agents, each object of the geographical data will be automatically used to instantiate an agent, GAMA taking care of managing the spatial projection of the data and, if necessary, of reading the values of the attributes.

In Figure 3, we present an extension of the model presented in Figure 2 in which *roads* and *buildings* agents are created from shapefile. Each object of the shapefile then becomes an agent: a *roads* agent for the road shapefile and a *buildings* agent for the building shapefile. At initialization, the *people* agents are located on one of the *building* agent randomly chosen.

GAMA provides as well many advanced spatial operators (spatial queries, union, difference, intersection....) that are directly usable through the GAML language. GAMA provides as well some high-level spatial operators : triangulation of a polygons, building of a graph from a set of points, polylines or polygons, skeletonization. At last, many movement operators and primitives are provided allowing to transparently move an agent on a grid, a graph or a continuous environment.

In Figure 4, we show an extension of the model of the Figure 3 in which a graph is created from the *roads* agents: each *roads* agent will become an edge of the graph. The *people* agents use this graph to move from building to building (a new target building is randomly chosen each time the people agent reaches the building).

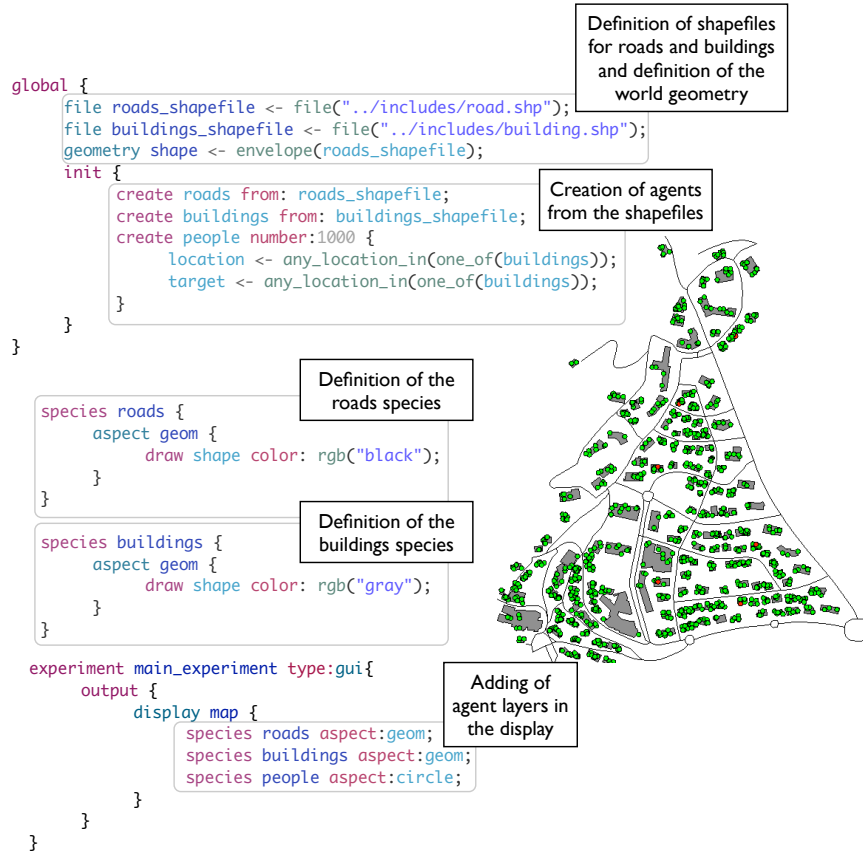


Fig. 3. Agentification of *roads* and *buildings* from shapefiles.

3.3 Visualization tools

With the advancement in computer graphics and the availability of large scale data, ABM community has a big role to play in a better understanding of complex systems by using high-level representation [6]. For instance, in large-scale agent-based simulation systems where virtual agents are situated in a virtual city [2]. Techniques available for visualization of model execution enhance the capacity to interpret, understand and explore a model, in particular to extract abstract data or discover imperceptible dynamics. Visualization is, in many cases, the only way to understand and study model that can not be expressed thanks to equations.

In already existing popular platforms, languages and tools for visualization do exist [24] but most of them lack techniques for building, observing and interacting with models. Those platforms stay focused on the agent representation

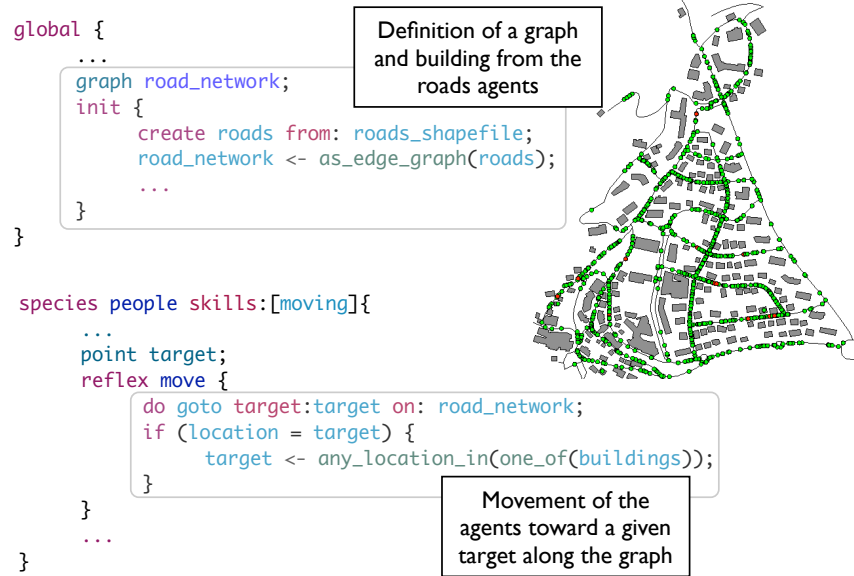


Fig. 4. Movement on a graph.

and do not propose any abstraction and multi-level representation or only by using ad-hoc solutions, loose coupling or post-simulation treatment that are not yet standardized. However, if some standardization categorization of ABM emerges and give guidelines for a better agent-based model representation [7], only few platforms propose built-in visualization tools that run online and serve as feedback or indicator during the simulation (e.g using graph analysis or data clustering to build groups of agents [17]). Actually, the 3D integration in agent-based model can be only achieve in two ways. The first one consists in a loose coupling between an agent-based platform with a 3D visualization toolkit. The second one is to build an agent-based models in 3D animation and rendering packages[6] where the model is developed within a specific environment like game engine to import and handle 3D data and agent behavior is defined by using other programming languages.

In GAMA, 3D is fully integrated to agent based modeling paradigm with a dedicated language that handle high-level representation [14]. The 3D visualization library in GAMA 1.6 uses OpenGL via the Java Binding for the OpenGL (JOGL) API library. GAMA offers a generic framework to separate the visual representation of the model from the underlying model, allowing different visual representations at different levels of abstraction, and insures that the visualization of a simulation and the interaction on a simulation are independent processes that do not alter the reference model itself [13]. In this approach a view is seen

as a model on which one can represent, abstract and interact without changing the definition of the observed model.

GAML language enables to define specific rich aspect by using operation such as repetition, modularity and recursion above the definition of graphical primitives. Agents are therefore associated with particular aspect and layers to determine its display location and representation. Agent can be displayed simultaneously in a variety of topologies (continuous, network, grid). GAMA allows the user to easily create a variety of geometric custom shapes, import and export scalable vector graphics-based geometry, images, GIS and 3D assets files. Geometrical operations can also be perform on the agent geometry to combined them by the the use of spatial operators (union, intersection, difference, scaling, rotation, etc.).

As mentioned in Section 3.2, GAMA allows a seamless integration of GIS data. Even with 2D GIS files, the handling of the third dimension is straightforward in GAMA where any depth can be assigned to a given shape. Thus, in Figure 5, we show a 3D rendering where a depth is added to the agent *buidlings* and where people agents are rendered as sphere instead of circle.

GAMA offers to modelers many tools to manipulate simulation displays. In particular, for each display, it provides a camera (Arcball and FreeFly) that can be used to reach any location of the model and therefore offers a more immersive experience. It also supports object selection, zooming, and panning. The new implementation of displays based on opengl has proven to be many times faster than the original Java2D approach when rendering large-scale simulations.

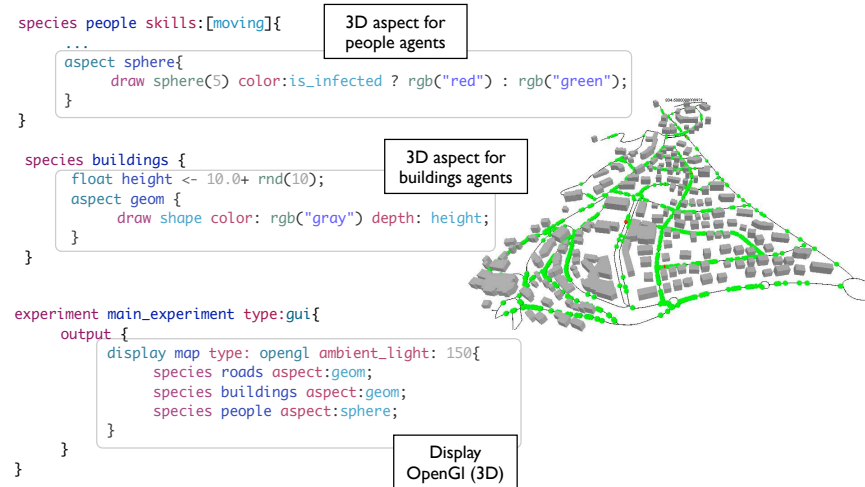


Fig. 5. Linking agents to places with geographic data. In GIS rendering, a depth is easily added to a shape file for a more realistic and accurate representation.

3.4 Multi-Level modeling

Multi-level agent-based modeling (ML-ABM) requires representing agents at different levels of representation in the same model with respect to time, space and behavior. Several reasons may justify the development of a multi-level agent-based. One reason may come from models that need to take into account entities at different spatial or organizational scales. For instance, models in biology may need to explicitly represent entities that belong to a hierarchy of containers, like molecules, cells and tissues. Another reason may be found in models in which the interactions between entities at one level make observables (usually called emergent structures [8]) appear, i.e. structures or patterns that are recognized as such by the modeler at another level of abstraction. In such a case, the modeler has the possibility to take into account and represent within agents, at the same time, the emergent structures and the agents that compose them. This problem of multi-level representation and articulation between agents at different levels is considered as one of the hardest issues to be tackled in the domain of complex systems modeling [29].

Recently, researchers of the community have begun to propose solutions to facilitate the development of multi-level agent-based models but existing approaches are still domain specific [25] [12] or only conceptual proposals without concrete implementation on established ABM platforms [21]. While current agent-based platforms (Repast [23], NetLogo [28]) still lack of appropriate abstractions, which explicitly support ML-ABM.

Thanks to the appropriate abstractions integrated in the meta-model, GAMA explicitly supports the development of multi-level agent-based model as shown in Figure 6. The following model illustrates how to write a multi-level agent-based model in GAML. Supposing that when people enter a building, the modeler would like to specify that the propagation of infection is controlled by the building basing on the number of infected people in the building. A possible solution is to specify that the people are modeled at two different levels of representation. The first representation level is when people are moving in the road network (represented by *people* species). The second representation level is when people are in the buildings. In this case, we define the *people.in_building* species (a sub-species of *people* species and a micro-species of *building* species). In this model, we unschedule all the agents of *people.in_building* species. When a *people* agent moves into a building, we use the *morphogenesis* operation (*capture* GAML statement), to change its species to *people.in_building* species. In this example, a *building* agent controls the infection propagation of its micro-agents using a simple mathematical formula. In real application, we can integrate a SIR model to the *building* agent in order to control the infection process. A person stays in a building for a certain amount of time. After that the *building* agent decides to release this person and the person continues to move in the road network. The modeler again uses the *morphogenesis* operation (*release* GAML statement), to change the representation level of people from *people.in_building* to *people*.

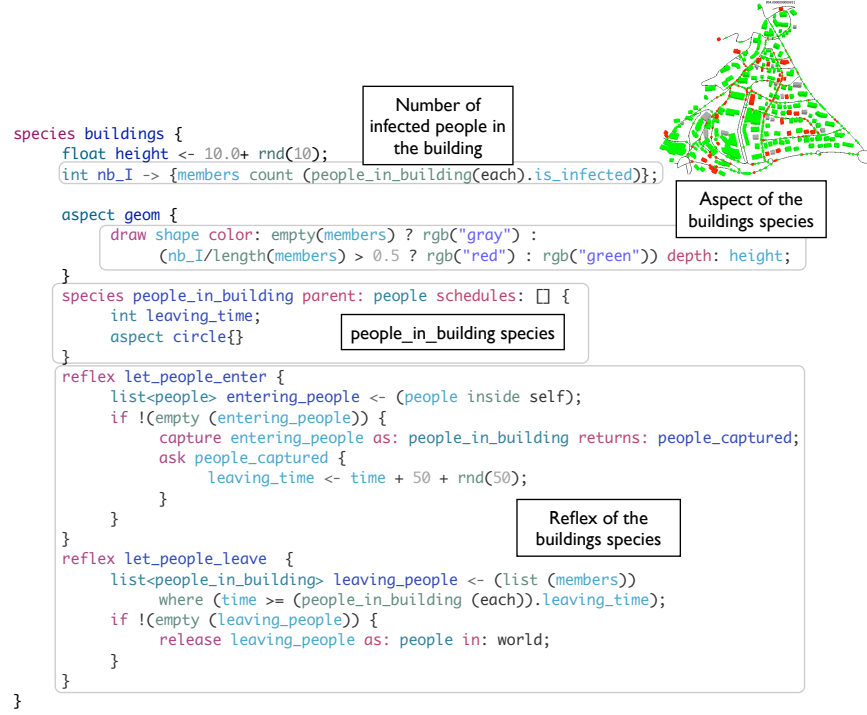


Fig. 6. Multi-level agent-based model in GAML

3.5 Equation-based modeling

Equation-based modeling (and in particular modeling using Ordinary Differential Equation, ODE) has been for a while the main modeling approach to represent dynamic systems. The most famous examples are the Lotka and Volterra [19] modeling of prey-predator dynamics or the Kermack and McKendrick [16] SIR model to represent epidemic dynamics.

Following the idea that we do not always need to represent every phenomenon at the individual level of details (as in the multi-level approach presented above) and the idea taken from meta-population trend in ecology [15], we have introduced in GAMA the possibility to describe the dynamics of agents variables using an ODE system and to integrate this system at each simulation step. A paradigmatic example of applications that would benefit from this new feature can take inspiration from Colizza et al. [5] or Meloni et al. [1] work. Let consider the network of main cities in the world connected by air flights. If a disease appears in a city it will quickly spread all around the world via these flights communication. In addition, in each city, the disease will also spread in the city population. In this case, we should consider thousands of millions of people. The observation level cannot be the individual one but rather the city one. These two

constraints induce the need but also the relevance of using an aggregate model at the city level where each city is represented by three variables representing the number of people in each compartment of the SIR model.

GAML enables to write equations linking agents attributes. Based on the Apache Commons Mathematics Library, we give the opportunity to the modeler to choose its integration method (between Runge Kutta 4 and Dormand-Prince 8(5,3)). The modeler can choose the integration step and the number of integration steps he wants to do in one simulation step. In Figure 7, we consider an epidemiological model at the level of a city. In this case, people are moving on a road network and buildings aggregate people. The dynamic in each building is managed using an SI equation system linking the number of Susceptible and Infected inhabitants located in the building.

This new way to describe the dynamics of agents will bring several major benefits. Modelers can dynamically switch during the simulation between agent-based approach and equation-based approach. In addition, we argue that the integration of equation-based model into agent-based platform will have pedagogical benefits not only for the GAMA platform but for all the modeling community. This will limit the entering cost for mathematicians into this new modeling approach, and will for sure induce a very interesting cross-fertilization between these two modeling approaches.

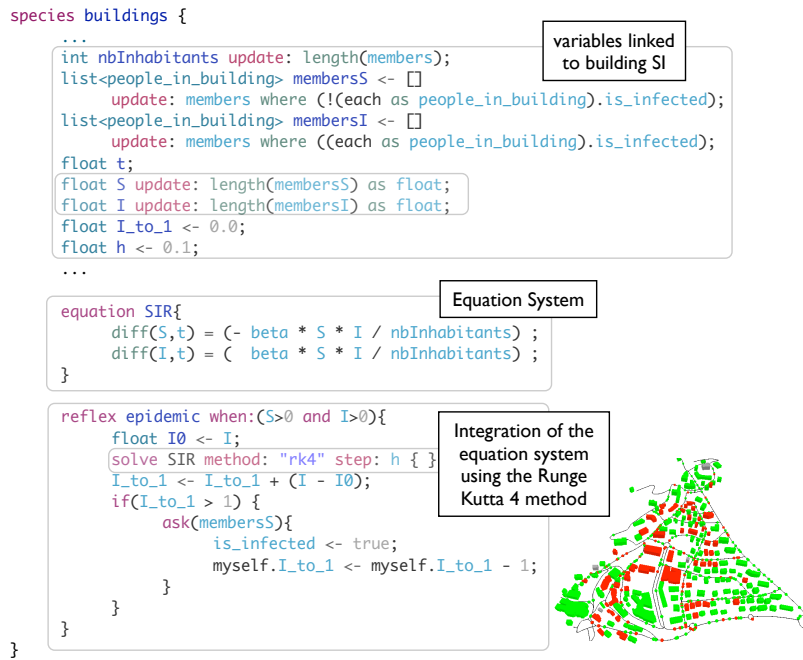


Fig. 7. Example of ODE-driven species.

4 Conclusion

Since the beginning of its development in 2007, GAMA has been deeply enriched. In particular, the new 1.6 version provides many powerful features to manage GIS data, advanced visualization and multi-level models. Many others features that were not presented in this paper are also available: database management (SQL), connexion to R, a 3D-physics engine plug-in. This increase of functionalities has allowed to gain an ever growing number of new users. Nowadays, GAMA is used for many ambitious research projects dealing with a wide variety of domains.

The first one is MAELIA project, which consists in the development of a decision-support tool for natural resource management [27] [11]. This project aims at studying the social, economic and ecological impact of water management in the Adour-Garonne Basin (France) by the integration a huge amount of geographical data and models (from ecological models to human decision-making models). The second one is the MIRO project [4] which addresses the issue of sustainable cities. Therefore, improving urban accessibility merely results in increasing the traffic and its negative externalities, while reducing the accessibility of people to the city. Given real data the simulator is used to realize scenarios determined by geographers for quantifying service accessibility and identifying cities management strategies. This model makes use of high resolution GIS data and large number of individuals. Another project concerns the simulation of evacuation organization in case of a tsunami in Nha Trang (Vietnam). This model combines detailed GIS data of the city and the coupling of agent and equation-based modeling in order to upscale the number of inhabitants that can be simulated [22].

In term of performance, experimental results show that the platform can easily handle several thousands of agents in real time rendering and reasonably deal with million of agents for non real time simulation.

GAMA is an ever-evolving platform and many enhancements are already planned for the future versions: a complete refactoring of the batch parameter space exploration, a plug-in to allow users to develop models through a graphical interface to modify the model in a more intuitive way than with the code, the possibility to call R scripts, the distribution of operators on a GPU, the possibility to share simulation where a model could be played at runtime on or in a replay mode at different spatio-temporal scale on different devices using WebGL to display simulation on a web browser. We did not mention it in this paper but we are also developing a 3D physics engine plug-in to obtain more realistic behavior in physical systems and primitives to access and interact with database management systems, in particular to store and retrieve spatial data.

References

1. Apolloni, A., Poletto, C., Colizza, V., et al.: Age-specific contacts and travel patterns in the spatial spread of 2009 h1n1 influenza pandemic. *BMC infectious diseases* 13(1), 1–18 (2013)
2. Araujo, F., Valente, J., Al-Zinati, M., Kuiper, D., Zalila-Wenkstern, R.: Divas 4.0: a framework for the development of situated multi-agent based simulation systems. In: *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. pp. 1351–1352. AAMAS '13, International Foundation for Autonomous Agents and Multiagent Systems (2013)
3. Axelrod, R.M.: *The complexity of cooperation: Agent-based models of competition and collaboration*. Princeton University Press (1997)
4. Banos, A., Marilleau, N.: Improving individual accessibility to the city: an agent-based modelling approach. *ECCS* (2012)
5. Colizza, V., Vespignani, A.: Epidemic modeling in metapopulation systems with heterogeneous coupling pattern: Theory and simulations. *Journal of theoretical biology* 251(3), 450–467 (2008)
6. Crooks, A.T., Castle, C.J.E.: *Agent-Based Models of Geographical Systems*. Springer Netherlands, Dordrecht (2012)
7. Daniel Kornhauser, U.W., Rand, W.: Design Guidelines for Agent Based Model Visualization. *Journal of Artificial Societies and Social Simulation* vol. 12, no. 2 1 12(2) (2009)
8. De Wolf, T., Holvoet, T.: Emergence versus self-organisation: Different concepts but promising when combined. In: *Engineering self-organising systems*, pp. 1–15. Springer (2005)
9. Drogoul, A., Amouroux, E., Caillou, P., Gaudou, B., Grignard, A., Marilleau, N., Taillandier, P., Vavasseur, M., Vo, D.A., Zucker, J.D.: Gama: multi-level and complex environment for agent-based models and simulations. pp. 1361–1362. AAMAS '13, International Foundation for Autonomous Agents and Multiagent Systems (2013)
10. Edmonds, B., Moss, S.: From kiss to kids—an anti-simplistic modelling approach. *Multi-Agent and Multi-Agent-Based Simulation* pp. 130–144 (2005)
11. Gaudou, B., Sibertin-Blanc, C., Thrond, O., Amblard, F., Arcangeli, J.P., Balestrat, M., Charron-Moirez, M.H., Gondet, E., Hong, Y., Louail, T., Mayor, E., Panzoli, D., Sauvage, S., Sanchez-Perez, J., Taillandier, P., Nguyen, V.B., Vavasseur, M., Mazzega, P.: The MAELIA multi-agent platform for integrated assessment of low-water management issues (regular paper). In: *International Workshop on Multi-Agent-Based Simulation (MABS)*, Saint-Paul, MN, USA. Springer Verlag (2013)
12. Gil-Quijano, J., Louail, T., Hutzler, G.: From biological to urban cells: lessons from three multilevel agent-based models. *Principles and Practice of Multi-Agent Systems* pp. 620–635 (2012)
13. Grignard, A., Drogoul, A., Zucker, J.D.: A model-view/controller approach to support visualization and online data analysis of agent-based simulation. In: *Proceedings of 2013 IEEE RIVF* (2013)
14. Grignard, A., Drogoul, A., Zucker, J.D.: Online analysis and visualization of agent based models. In: *Computational Science and Its Applications-ICCSA 2013*, pp. 662–672. Springer (2013)
15. Hanski, I.: *Metapopulation Ecology*. Oxford University Press (1999)

16. Kermack, W.O., McKendrick, A.G.: A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London* 115(772), 700–721 (1927)
17. Lamarche-perrin, R., Demazeau, Y., Vincent, J.m.: How to Build the Best Macroscopic Description of your Multi-agent System ? pp. 1–18 (2013)
18. Le Page, C., Bousquet, F., Bakam, I., Bah, A., Baron, C.: Cormas: A multiagent simulation toolkit to model natural and social dynamics at multiple scales. In: *Proceedings of Workshop "The ecology of scales"*, Wageningen (The Netherlands) (2000)
19. Lotka, A.J.: Contribution to the theory of periodic reactions. *The Journal of Physical Chemistry* 14(3), 271–274 (1909)
20. Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., Balan, G.: Mason: A multiagent simulation environment. *Simulation* 81(7), 517–527 (2005)
21. Morvan, G.: Multi-level agent-based modeling-bibliography. *arXiv preprint arXiv:1205.0561* (2013)
22. Nguyen, T.N.A., Zucker, J.D., Nguyen, H.D., Drogoul, A., Vo, D.A.: A hybrid macro-micro pedestrians evacuation model to speed up simulation in road networks. In: *Advanced Agent Technology*, pp. 371–383. Springer (2012)
23. North, M.J., Collier, N.T., Ozik, J., Tatara, E.R., Macal, C.M., Bragen, M., Sydelko, P.: Complex adaptive systems modeling with repast symphony. *Complex Adaptive Systems Modeling* 1(1), 1–26 (2013)
24. Railsback, S.F., Lytinen, S.L., Jackson, S.K.: Agent-based Simulation Platforms: Review and Development Recommendations. *Simulation* pp. 609–623 (2006)
25. Servat, D., Perrier, E., Treuil, J.P., Drogoul, A.: When agents emerge from agents: Introducing multi-scale viewpoints in multi-agent simulations. In: *Multi-Agent Systems and Agent-Based Simulation*. pp. 55–61. Springer (1998)
26. Taillandier, P., Amouroux, E., Drogoul, A.: GAMA : a simulation platform that integrates geographical information data , agent-based modeling and multi-scale control. *Principles and Practice of Multi-Agent Systems and Writing* (13th International Conference, PRIMA 2010, Kolkata, India, November 12-15, 2010), 67–74 (2010)
27. Taillandier, P., Thrond, O., Gaudou, B.: A new BDI agent architecture based on the belief theory. Application to the modelling of cropping plan decision-making (regular paper). In: *International Environmental Modelling and Software Society (iEMSs)*, Leipzig, Germany, 01/07/2012-05/07/2012 (2012)
28. Tisue, S., Wilensky, U.: Netlogo: A simple environment for modeling complexity. In: *International Conference on Complex Systems*. pp. 16–21 (2004)
29. Treuil, J.P., Drogoul, A., Zucker, J.D.: *Modélisation et simulation à base d'agents: exemples commentés, outils informatiques et questions théoriques*. Dunod (2008)